

REMARKS

This paper addresses the office action dated February 11, 2004. Claims 1-46 stand rejected. New claims 47-55 have been added. The amendment is supported throughout the originally filed disclosure. No new matter has been added by the present amendment. The new claims are new, nonobvious, and useful.

Claims 1-55 are presently pending. Applicants respectfully request reconsideration of the existing claims and examination of the new claims in light of the remarks given below.

REJECTION OF CLAIMS 1-5, 13-18, 26-28, 35-38, 45-46 UNDER 35 U.S.C. §102

Claims 1-5, 13-18, 26-28, 35-38, 45-46 stand rejected under 35 U.S.C. § 102(e) in light of U.S. Patent No. 5,953,532 to Lochbaum ("Lochbaum"). Applicants traverse the rejection for at least the reasons given below.

Some embodiments of the present invention are generally directed to a system and method for creating downloadable applications and bootable applications on a development system for download to a target system, e.g., an embedded processor. The bootable applications may include various operating system loads for the target system. In particular, these applications may include domains which comprise various types of components. Components of the downloadable application or bootable application may include software modules, e.g., executable modules, a symbol list for linking, constraints, parameters, and parameter values.

Some embodiments of the system may also facilitate dynamic linking of new downloadable applications and bootable applications on to a running target system. Symbols may represent particular memory or code locations. Symbols may be imported into modules, exported by a module, or private to a module. In particular some symbols may correspond to entry points for domains.

In contrast, Lochbaum generally describes a system for administering a system of networked PCs. Lochbaum allows administrators or users to download new applications to PCs when they are requested by a user. As will be discussed in more detail below, Lochbaum does not address the particular aspects of Applicants' claimed invention such as symbols, domains, and component access. Rather, Lochbaum addresses user-level applications and system administration. Applicants respectfully submit that Lochbaum does not anticipate Applicants' claim 1, for at least the reason that several recited features of claim 1 are missing from Lochbaum.

More particularly, Claim 1 recites:

1. A method, comprising:
 - (1) determining a set of present components assigned to a domain, each of the set of present components includes a set of modules;
 - (2) determining a set of symbols imported by the set of modules assigned to the domain;
 - (3) determining zero or more needed components to which the domain does not have access and at least one of provides the set of symbols imported by the set of modules, and specified as required by the set of present components; and
 - (4) adding the zero or more needed components into the domain.

As an initial matter, the Examiner indicates that Lochbaum's second child window 220 and col. 4:7-10 meet the recited limitation of Applicants' claim 1 "determining a set of symbols imported by the set of modules assigned to the domain". Applicants respectfully traverse. Lochbaum's 4:7-10 merely describes a list of application programs which can be downloaded to a PC. Lochbaum neither teaches nor suggests determining a set of imported symbols, let alone a set of imported symbols assigned to a domain which has had a present component analysis performed upon it. An examination of Applicants' specification make it clear that "symbol" is used as is typical in the linking and compiling arts. Moreover, even if the listed applications in Lochbaum were "symbols", as recited in claim 1, there is no teaching or suggestion in Lochbaum of determining symbols that are "imported by the set of modules assigned to the domain".

This determination is entirely absent from Lochbaum. This is unsurprising, because, as described above, Lochbaum is in quite a different art than Applicants' claimed invention, which deals with development environments rather than client-server system administration.

Furthermore, Applicants' claim 1 recites

“determining zero or more needed components to which the domain does not have access and at least one of provides the set of symbols imported by the set of modules, and specified as required by the set of present components”.

The Examiner points to Lochbaum 4:8-10 for this feature. Indeed, this portion of Lochbaum refers to applications to which a user does not have access. But at that point Lochbaum is finished – the user simply can not have access to the missing application. In contrast in the system claimed by Applicants' claim 1, the components to which the domain does not have access may be represented by a set of imported symbols, e.g., cross-references to entry points in other domains which do have access to these components. These needed components are then added in the final portion of Applicants' claim 1, “adding the zero or more needed components into the domain”. The added needed components may include cross reference to components to which the domain does not have access. The Applicants' analysis to determine and obtain missing components is entirely different from the pure permission checking approach of Lochbaum.

As an example, the domain for which components are being assigned might need access to a protected device, e.g., an I/O device such as a tape drive. A conventional user process would probably not be allowed to directly access this device. For example, an I/O device like a tape drive might be only accessible to one process at a time and the special domain might provide some sort of mutual exclusion control to give sequential access to the drive to different processes. The needed component might be found in the special domain which controls special critical resources that need protection, e.g., the tape drive. This special domain which provides tape drive access would need to be added to a build for a user process which needs access to the tape drive be; it is a needed component but it is not part of the domain and the domain might not have direct access to it. In contrast, as

claimed in other claims, if no process uses the tape drive, the special domain might be omitted from a system build.

Because several elements of Applicants' claim 1 are not found in the cited Lochbaum reference, Applicants respectfully submit that Lochbaum does not anticipate Applicants' claim 1. Withdrawal of the rejection of claim 1 under 35 U.S.C. § 102 is respectfully requested.

Claims 2-5 depend from claim 1 and thus should be patentable for at least the reasons given above for claim 1.

Claim 13 should be patentable for at least reasons similar to those given above for claim 1. Moreover, claim 13 recites a parser "that lists a set of symbols and a set of modules that imports or exports the set of symbols". Claim 13 also recites "a second parser that maps the set of modules to a set of components and specifies dependencies among the set of components" and a "project analysis utility." The Examiner has not identified these features in Lochbaum. Accordingly, Applicants respectfully submit that Lochbaum does not anticipate claim 13, and respectfully request withdrawal of the rejection

Claims 14-18 depend from claim 13, and thus should be patentable over Lochbaum for at least all the reasons given above for claim 13. Claim 15 should also be patentable for reasons similar to those given above for claim 1, where the Applicants discuss the difference between the recited claim language and the access control elements of Lochbaum.

Claim 26 recites in part

determining a set of symbols imported by the set of modules in each of the zero or more precious components.

As was discussed above for claim 1, Applicants respectfully submit that this determining a set of imported symbols is neither taught nor suggested by Lochbaum. Similarly claim 26 also recites

Determining zero or more needed components to which the domain does not have access and at least one of provides the set of symbols imported by the set of modules, specified as required by the zero or more previous components

As was discussed above for claim 1, this language is quite different than the conventional access control described in Lochbaum. In particular, if the domain does not have access to the needed components and they are found in the present components, they will be moved to the previous components. Applicants respectfully submit that this limitation is not taught or suggested by Lochbaum.

Because neither of the recited limitations are taught or suggested by Lochbaum, Applicants respectfully submit that Lochbaum does not anticipate claim 26. Therefore, withdrawal of the rejection of claim 26 is respectfully requested.

Claims 27 and 28 depend from claim 26, and thus should be patentable for at least the reasons given above for claim 26.

Claim 35 recites two parsers, a first parser that lists a set of symbols and a set of modules that imports or exports the set of symbols, and a second parser that maps the set of modules to a set of components and specifies dependencies. The Examiner has not identified these recited elements in Lochbaum. The Applicants respectfully submit that Lochbaum does not teach or suggest these limitations, and therefore does not anticipate claim 35. For at least this reason, Applicants respectfully request withdrawal of the rejection of claim 35.

Claims 36-38 depend from claim 35 and thus should be patentable for at least the reasons given above for claim 35.

The Examiner indicates that claims 45 and 46 correspond to claims 1 and 26. They thus should be patentable for at least the reasons given above for claims 1 and 26.

REJECTION OF CLAIMS 6-12, 19-25, 29-34, AND 39-44 UNDER 35 U.S.C. § 103(A)

Claims 6-12, 19-25, 29-34, and 39-44 stand rejected under 35 U.S.C. § 103(a) over Lochbaum in light U.S. Patent No. 5,528,757 to Yamasaki (“Yamasaki”). All of these claims depend, directly or indirectly from claims discussed previously. They thus should be patentable for at least the reasons given above for their parent claims.

Next, the Applicants respectfully traverse the Examiner’s suggestion that Yamasaki and Lochbaum are analogous art. To rely on a reference under 35 U.S.C. § 103, it must be in an analogous art. *See* MPEP 2141.01. Neither cited reference deals with a development environment for domains, downloadable applications, or bootable applications on a target system. Rather, both deal primarily with network administration for client-server user computers. Because neither reference is analogous, Applicants’ respectfully submit their proposed combination does not render any of Applicants’ claims obvious.

Applicants also respectfully traverse the proposed combination of Lochbaum and Yamasaki because the Examiner has not provided a proper motivation to combine. Rather, the Examiner appears to pull his proposed reasons for making the combinations from thin air, strong evidence of an improper hindsight reconstruction. To reject, the Examiner must find a suggestion to combine the references that is “clear and particular”. *In re Dembiczak*, 50 U.S.P.Q.2d 1614 (Fed. Cir. 1999). The requirement is for “actual evidence” of the proposed motivation to combine. *Teleflex, Inc. v. Ficosa North America Corp.*, 299 F.3d 1313, 1334, 63 USPQ2d 1374 (Fed. Cir. 2002). The fact that references can be combined or modified does not render the resultant combination or modification obvious unless the prior art also suggests the desirability of the combination or modification. *See* MPEP 2143.01 (citing *In re Mills*, 916 F.2d 680 (Fed. Cir. 1990)). Because there is no particular suggestion from a cited reference, or an affidavit of official

notice, Applicants' respectfully submit the Examiner's combination is improper and respectfully request withdrawal of the section 103 rejections.

Furthermore, Claim 6 recites "a domain link path". The domain link path is a hierarchy or map which controls the linking of a group of components into an executable module. For example, UNIX makefiles are sometimes used to define the linking of UNIX object files. The feature identified by the Examiner in Yamasaki is merely a network connection – it has nothing to do with program linking and compilation.

With respect to claim 7, the Examiner identifies the physical layer of an OSI connection as "the kernel layer". Although the "kernel", a well-known concept in the operating system art, provides an interface to the hardware and to other processes in the system, it does not correspond and can not be equated with the "physical layer" of an OSI connection.

For at least all the above reasons, neither Lochbaum, Yamasaki, nor their combination renders claims 6-12, 19-25, 29-34, or 39-44 obvious. Withdrawal of the rejection under 35 U.S.C. § 103 is respectfully requested.

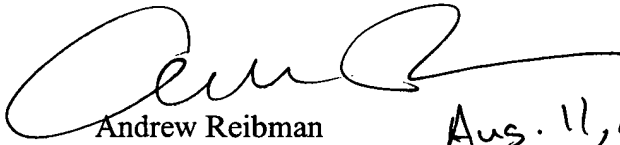
NEW CLAIMS

New claims 47-55 have been added to recite further features of Applicants' invention. These claims depend from claims 1 and 13 and thus should be patentable for at least the reasons given above for parent claims 1 and 13.

CONCLUSION

All rejections having been addressed, Applicants' respectfully request withdrawal of the rejections, examination of the newly added claims, and prompt passage to allowance.

Respectfully submitted,
KENYON & KENYON


Andrew Reibman
(Reg. No. 47,893)

Aug. 11, 2004

One Broadway
New York, NY 10004
Phone: (212) 425-7200
Fax: (212) 425-5288

CUSTOMER NO. 26646